2006 IEEE Congress on Evolutionary Computation
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

PESO+ for Constrained Optimization

Angel E. Muñoz-Zavala, Arturo Hernández-Aguirre, Enrique R. Villa-Diharce and Salvador Botello-Rionda

***Abstract*— We introduce the PESO+ algorithm (Particle Evolutionary Swarm Optimization Plus) for the solution of single objective constrained optimization problems. A novel feature introduced by PESO+ is an external archive to store and retrieve "tolerant" particles found at past tolerance values. This technique is aimed to preserve particles that otherwise would be lost after the adjustment of the tolerance of equality constraints. Also, two perturbation operators, "c-perturbation" and "m-perturbation" are described. The goal of these operators is to keep diversity and to prevent premature convergence. The constraint handling technique is based on feasibility and summation of constraint violations. All experimental results are reported as required by the organizers of the special session on "Constrained Real Parameter Optimization" at CEC2006.**

I. Introduction

The PSO (Particle Swarm Optimization) algorithm takes inspiration from the motion of a flock of birds [5]. The whole flock moves following its leader but the leadership can be passed from member to member. At every PSO iteration the flock is tested for a new leader: if a member is found to improve the current leader for guiding the flock to a more promissory place, then that member takes the leadership. In this model, the new position of every member of the flock is computed by combining its local information and the position of the current leader.

Two issues arise when PSO is applied to constrained optimization problems. The first is how fast the flock members should follow the best. Should they move slowly towards the leader so on the way they can hope for a better option?. The situation gets more complicated when the leader approaches a feasible region. The additional pressure on the flock (since no other member will do better), may cause the flock to miss the optimum due to premature convergence. Similar convergence problems have been reported elsewhere [7]. The second issue is related to the information collected by the flock. All flock members watch for promissory areas and every one updates its best visited location in a local store. Since the computation of a new leader is strongly biased by past collective experiences learnt by the flock, the member with the best visited location in store becomes the new leader. PESO+ will try to improve the knowledge about the position of visited locations (stored in PBest), thus the next leader may guide the flock to a better place.

PESO+ algorithm introduces two main features that take care of the noted issues: 1) two perturbation operators aimed to improve the collected knowledge so the new leader is more reliable in guiding the flock; 2) an external file (called TF) to preserve feasible individuals at different tolerance values of equality constraints. Since the tolerance decreases from generation to generation, the filed particles can be inspected and the more promissory retrieved and inserted in the flock.

This paper is organized as follows: Section I provides the definition of the class of problem we wish to solve; Section III is dedicated to explain the PESO+ algorithm in detail; the results of all experiments are reported in Section IV, and conclusions are provided in Section V.

Angel E. Muñoz-Zavala is a Ph.D. student of Computer Science, Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Gto 36000, Mexico (email: aemz@cimat.mx).

Arturo Hernández-Aguirre is with the Department of Computer Science, Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Gto 36000, Mexico (email: artha@cimat.mx).

Enrique R. Villa-Diharce is with the Department of Probability and Statistic, Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Gto 36000, Mexico (email: villadi@cimat.mx).

Salvador Botello-Rionda is with the Department of Computer Science, Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Gto 36000, Mexico (email: botello@cimat.mx).

II. Problem Statement

We are interested in the general nonlinear programming problem in which we want to:

$$\text{Find } \ \vec{x} \text{ which optimizes } f(\vec{x})$$

subject to:

$$g_i(\vec{x}) \leq 0, \quad i = 1, \ldots, n$$

$$h_j(\vec{x}) = 0, \quad j = 1, \ldots, p$$

where $\vec{x}$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_r]^T$, n is the number of inequality constraints and p is the number of equality constraints (in both cases, constraints could be linear or non-linear). For an inequality constraint that satisfies $g_i(\vec{x}) = 0$, then we will say that is active at $\vec{x}$. All equality constraints h_j (regardless of the value of $\vec{x}$ used) are considered active at all points of $\mathcal{F}$ ($\mathcal{F}$ = feasible region).

III. The PESO+ algorithm

PESO+ implements the standard PSO heuristic but this is improved with external procedures that do not affect the members of the flock. The main feature that drives PSO is social interaction and the social psychological tendency of individuals to follow the success of the best member of the flock. Thus, in PESO+, only the memory of best locations (PBest) is altered by perturbation operators. PBest can be updated with improved locations but the actual particles are never modified. The flock will follow the improved PBest at the next turn to fly. A bird's eye view of PESO+ is shown in Figure 1 depicting the main data flow. Note that on stages two and three, just after a perturbation operation and also after a call to Best-tolerant, PBest is the only file updated (all details are given in Section III-G).

0-7803-9487-9/06/$20.00/©2006 IEEE

```
p = 1.0
ε = 1.0
Do
   Stage 1
   Pbest ← call(PSO)
   Stage 2
   If p ≤ U(0,1) then
      Pbest ← C-Perturbation(Pbest)
      Pbest ← Best-tolerant(Pbest, TF)
   Stage 3
   If p ≤ U(0,1) then
      Pbest ← M-Perturbation(Pbest)
      Pbest ← Best-tolerant(Pbest, TF)
p = p − δ
ε = ε − μ
Until termination criteria is met
```

Fig. 1. High level view of PESO+

A. The standard PSO algorithm

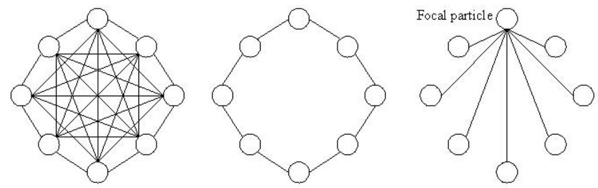
The PSO algorithm is the main actor of PESO+. Following the standard PSO implementation [2], the *velocity* vector is computed from $PBest$ and $LBest$ and later used to update the particles position. Pbest is a memory of the best position of every particle. Lbest is also a memory of the best position of a particle in a neighborhood of size $n = 2$. The velocity vector drives the optimization process and reflects the socially exchanged information. Figure 2 shows the pseudo-code of **Velocity** function, where n is the population size, d is the dimension of the search space, $c1$ and $c2$ are constants set to 1, and w is the inertia weight whose random value comes from a uniform distribution in $[0.5, 1]$. The inertia weight controls the influence of previous velocities on the new velocity.

```
For k = 0 To n
   For j = 0 To d
      r1 = c1 * U(0, 1)
      r2 = c2 * U(0, 1)
      w = U(0.5, 1)
      V_{i+1}[k, j] = w * V_{i+1}[k, j] + r1 * (PBest_i[k, j] - P_i[k, j]) +
                     r2 * (PBest_i[LBest_i(k), j] - P_i[k, j])
   End For
End For
```

Fig. 2. Pseudo-code of **Velocity**(V_i, P_i, $PBest_i$, $LBest_i$)

B. Topological organization

PESO+ best members are determined by the "local best" criteria (Lbest) instead of the popular global best. Lbest-PSO has been reported to traverse larger areas of the search space, hence favoring exploration and preventing premature convergence (slower convergence speed has been reported [2], [6]). Lbest works on neighborhoods of particles [4], some of the most common organizations are shown in Figure 3. In PESO+, neighborhoods are structured in a ring fashion, as shown in Figure 3(b). Flock members organized in ring communicate with n immediate neighbors, $n/2$ on each side. Every particle is initialized with a permanent label so the local best neighbor of particle i is determined by testing particles $i + 1$ and $i - 1$, independently of their geographic location.



(a) Star Neighborhood Structure (b) Ring Neighborhood Structure (c) Wheel Neighborhood Structure

Fig. 3. Neighborhood structures for PSO

C. Constraint handling rules

The local best neighbor of the i_{th} particle is found by applying a set of well known rules called "feasibility and summation of constraint violations" [8]. These rules are: 1) given two feasible particles, pick the one with better fitness value; 2) if both particles are infeasible, pick the particle with the lowest sum of constraint violation, and 3) from a pair of feasible and infeasible particles, pick the feasible one. These rules are used by PESO+ every time the "best" members of a set need to be computed, for instance, for the particle best set, $PBest$.

D. Dynamic tolerance for equality constraints

PESO+ handles equality constraints by rewriting them as inequality constraints of the form $|h_j| \leq \epsilon$, where ϵ is called the tolerance. In PESO+ the tolerance linearly decreases from 1.0 to the target value of 0.0001 during the first 90% of the featness evaluations. For the last 10% the tolerance is kept fixed.

E. PSO + Perturbations + Constraint Handling = PESO+

The PESO+ algorithm does use two perturbation operators to keep diversity and exploration. Figure 1 shows the three stages of PESO+: in the first stage the standard PSO algorithm is performed [6], then the perturbations are applied to $PBest$ in the next two stages. The goal of the second stage is to add a perturbation generated from the linear combination of three random vectors, in a way similar to the "reproduction operator" found in *differential evolution* algorithm [10]. Similar approaches try to improve PSO by applying perturbations to the flock itself. However, the PSO behavior is altered and the success is limited [11]. In PESO+ this perturbation is called C-Perturbation. It is applied to the members of PBest to yield a set of temporal particles $Temp$. Then each member of $Temp$ is compared with its corresponding father and PBest is updated with the child if it wins the tournament. Figure 4 shows the pseudo-code of the **C-Perturbation** operator, where r is a random uniform number between 0 and 1.

In the third stage every vector is perturbed again so a particle could be deviated from its current direction as responding to external, maybe more promissory, stimuli. This perturbation is implemented by adding small random numbers with uniform distribution, a kind of small noise. The perturbation, called M-Perturbation, is applied to every member of PBest to yield a set of temporal particles $Temp$.

```
For k = 0 To n
  For j = 0 To d
    r = U(0, 1)
    p1 = k
    p2 = Random(n)
    p3 = Random(n)
    Temp[k, j] = P_{i+1}[p1, j] + r (P_{i+1}[p2, j] - P_{i+1}[p3, j])
  End For
End For
```

Fig. 4. Pseudo-code of **C-Perturbation**(P_{i+1})

Then each member of $Temp$ is compared with its corresponding father and PBest is updated with the child if it wins the tournament. Figure 5 shows the pseudo-code of the **M-Perturbation** operator. The perturbation is added to every dimension of the decision vector with probability $p = 1/d$ (d is the dimension of the decision variable vector). LL and UL are the lower and upper limits of the search space.

```
For k = 0 To n
  For j = 0 To d
    r = U(0, 1)
    If r ≤ 1/d Then
      Temp[k, j] = Rand(LL, UL)
    Else
      Temp[k, j] = P_{i+1}[k, j]
  End For
End For
```

Fig. 5. Pseudo-code of **M-Perturbation**(P_{i+1})

These perturbations have the additional advantage of keeping the self-organization potential of the flock since they only work on the PBest memory. Note that these perturbations are also suitable for real-valued function optimization.

F. Refining Solutions

The cooperation between PSO and the perturbation operators have been studied by the authors who have carefully analyzed the data from the many experiments conducted. The PSO section is quite efficient at refining solutions in a local subspace but the new perturbation operators greatly help the flock to improve their performance at discovering promissory local subspaces (where a potential solution resides). Hence, the perturbation operators must greatly explore the space during the first generations but their strength should be decremented so a refining phase conducted by PSO may take place. The implementation of these cooperative activities is as follows: the perturbation operators are applied to PBest with probability $p = 1$ when the flock is flown for the first time. This probability is constantly and linearly decremented reaching its final value of $p = 0$ at the last time the particles fly.

G. Diversity control

PESO+ performs two specific tasks oriented to enhance the overall population diversity but also near the boundary of equality constraints.

1) Keeping infeasible particles in the flock. A few unfeasible particles are kept in the flock even when most particles might already be inside the feasible region. This is the way PESO+ diminishes the pressure on the flock to enter the feasible region, thus preserving exploration during more generations. Every time Pbest memory is updated with the best perturbed particles, a random subset of 80% of the flock is tested by feasibility and summation of constraint violations criteria. The remaining 20% is determined by comparing fitness values. These actions are undertaken with some probability p. In PESO+, the initial value of $p = 0.5$ varies linearly through generations until it gets $p = 0.0$ in the last generation. (this powerful idea about preserving infeasible individuals was introduced by the Stochastic Ranking algorithm [9]).
2) Storage of tolerant particles. As it is customary for evolutionary algorithms, PESO+ rewrites equality constraints as inequalities plus a tolerance value named ϵ. Thus we get the final form ($|h_j| \leq \epsilon$). We call *tolerant* a particle that remains feasible after two or more consecutive reductions of that tolerance. We have explained in Section III-D that the tolerance is dynamic, having its larger value the first time the particles fly. A tolerant particle can survive several tolerance levels because it may find a high quality position *near* the equality constraints boundary. Thus, it remains feasible after several decrements of the tolerance. Some others are extinguished after the next decrement of the tolerance value. PESO+ keeps in a file the tolerant particles so every time the perturbations are applied to PBest and the tolerance has been decremented, the best particle of PBest, say a, is inserted into the *tolerant file*. Next all particles in the tolerant file have their equality constraints evaluated and the one with the less amount of violation is retrieved and inserted in PBest replacing particle a.

Although dynamic constraint levels have been used for long time (received major attention again when the ASCHEA algorithm promoted its use [3]), the tolerant file is a unique feature of PESO+. The pseudo-code of Best-tolerant is shown in Figure 6.

```
Procedure Best-tolerant(PBest)
% TF is the file of tolerant particles, initially empty
q ← bestparticle(PBest)
if TF == full
  TF ← TF \ any one in TF
TF ← TF ∪ q
TF ← computeconstraints(TF)
bp ← bestparticle(TF)
PBest ← PBest \ q
PBest ← PBest ∪ bp
```

Fig. 6. Procedure Best-tolerant of PESO+

For PESO+ the total amount of violation measured on the equality constraints helps to determine a better leader for the flock. Other approaches simply consider that any individual is as good as any other if their amount of violation on each equality constraint is less than the tolerance ϵ.

H. Pseudocode of PESO+ algorithm

In Figure 7 the main algorithm of PESO+ is listed. TF is the tolerant file (for problems with equality constraints), updated immediately after a perturbation takes place. p is a linearly decreasing probability from 1.0 to 0 (according to the function evaluations), and k represents the $PBest$ individual that has the best solution at the current generation.

```
PBest_0 = P_0 = Rand(LL, UL)
FBest_0 = F_0 = Fitness ( P_0 )
CBest_0 = C_0 = SCV ( P_0 ) % SUM OF CONSTRAINT VIOLATIONS
V_0 = Rand(-(UL-LL), (UL-LL))
TF = < PBest_0[k], FBest_0[k], CBest_0[k] >
% TOLERANT FILE
Do
  LBest_i = LocalBest ( FBest_i, CBest_i )
  V_{i+1} = Velocity ( V_i, P_i, PBest_i, LBest_i)
  P_{i+1} = P_i + V_{i+1}
  F_{i+1} = Fitness ( P_{i+1} )
  C_{i+1} = SCV ( P_{i+1} )
  < PBest_{i+1}, FBest_{i+1}, CBest_{i+1} > =
    Best ( PBest_i, P_{i+1}, FBest_i, F_{i+1}, CBest_i, C_{i+1} )
  TF = TF ∪ < PBest_{i+1}[k], FBest_{i+1}[k], CBest_{i+1}[k] >
  < PBest_{i+1}[k], FBest_{i+1}[k], CBest_{i+1}[k] > = Best ( TF )
  If( U(0,1) < p )
    Temp = C-Perturbation (P_{i+1} )
    FTemp = Fitness ( Temp )
    CTemp = SCV ( Temp )
    < PBest_{i+1}, FBest_{i+1}, CBest_{i+1} > =
      Best ( PBest_i, Temp, FBest_i, FTemp, CBest_i, CTemp )
    TF = TF ∪ < PBest_{i+1}[k], FBest_{i+1}[k], CBest_{i+1}[k] >
    < PBest_{i+1}[k], FBest_{i+1}[k], CBest_{i+1}[k] > = Best ( TF )
  End If
  If( U(0,1) < p )
    Temp = M-Perturbation (P_{i+1} )
    FTemp = Fitness ( Temp )
    CTemp = SCV( Temp )
    < PBest_{i+1}, FBest_{i+1}, CBest_{i+1} > =
      Best ( PBest_i, Temp, FBest_i, FTemp, CBest_i, CTemp )
    TF = TF ∪ < PBest_{i+1}[k], FBest_{i+1}[k], CBest_{i+1}[k] >
    < PBest_{i+1}[k], FBest_{i+1}[k], CBest_{i+1}[k] > = Best ( TF )
  End If
End Do
```

Fig. 7. Main Algorithm of $PESO+$

IV. Experiments

In this section we present the results of all experiments proposed by the Special Session on Constrained Real Parameter Optimization [1].

PC Configuration. System: Windows XP; CPU: Pentium 4 at 3.00GHz; RAM: 1.00 GB; Language: C++ Builder; Algorithm: PESO+

Parameters Setting. PESO+ is not sensible to factors $c1$ and $c2$ so they are set to 1. W is a random number in the interval [0.5,1] with uniform distribution. Perturbations are performed with some probability as explained in Section III-D. The population size is set to 100 for all problems.

Algorithm Complexity. In Table I we show the complexity of the algorithm. The elapsed times $T1$ and $T2$ were measured for 10000 evaluations (function and constraints) for each problem, using a population of 100 particles. $T1$ represents the average computing time of 10000 evaluations for each test problem; and $T2$ is the average computing time of 10000 evaluations for all benchmark problems using the PESO+ algorithm.

TABLE I
Computational Complexity

$T1$	$T2$	$(T2-T1)/T1$
0.4421	0.5538	0.2527

Experimental results. In Tables II, III, IV and V we present the best, median, worst, mean and standard deviation of the error values achieved for each test problem $(f(x)-f(x*))$; where c is the number of violated constraints at the median solution: the sequence of three numbers indicate the number of violations (including inequality and equalities) by more than 1.0, more than 0.01 and more than 0.0001 respectively. v is the mean value of the violations of all constraints at the median solution. The numbers in the parenthesis after the fitness value of the best, median, worst solution are the number of constraints which can not satisfy feasible condition at the best, median and worst solutions respectively.

In Table VI, we present the number of fitness function evaluations to achieve the required accuracy level $((f(\vec{x}) - f(\vec{x}^*)) \leq 0.0001$. Therefore, it only shows the measures on the successful runs. The number of successful runs is shown in the column *Success Rate*. The *Success Performance* is the result of $\frac{Totalruns \times Mean}{Successfulruns}$.

Convergence Map. Convergence maps are shown in Figures 8, 9, 10 and 11. Convergence maps plot the median $log_1 0(f(x) - f(x^*))$ or $log_1 0(v)$ values for the 25 runs. Points which satisfy $f(x) - f(x^*) \leq 0$ or $v \leq 0$ are not plotted.

Results of PESO+. Table VII shows the results of PESO+ for the benchmark problems in the common format used by the evolutionary constrained optimization community.

V. Conclusions and Future Work

PESO+ is a standard PSO algorithm enhanced with two perturbation operators that do not destroy the flock concept neither its self-organization capacity. PESO+ is simple and easy to implement. Besides, it only needs the tuning of very few control and mutation parameters. The additional external file to store the real amount of equality constraint violation of any particle is a unique feature that PESO+ promotes. The results on the benchmark prove that PESO+ is highly competitive.

References

[1] *http://www.ntu.edu.sg/home/EPNSugan*.

[2] R. Eberhart, R. Dobbins, and P. Simpson. *Computational Intelligence PC Tools*. Academic Press Professional, 1996.

[3] S. B. Hamida and M. Schoenauer. Aschea: new results using adaptive segregational constraint handling. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC-2002*, pages 884 – 889. IEEE, May 2002.

[4] J. Kennedy. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In *IEEE Congress on Evolutionary Computation*, pages 1931–1938. IEEE, July 1999.

TABLE II

ERROR VALUES ACHIEVED WHEN FES= 5×10^3 , FES= 5×10^4, FES= 5×10^5 FOR PROBLEMS 1-6.

FES		g01	g02	g03	g04	g05	g06
5×10^3	Best	9.1901 (0)	4.3547E-01 (0)	-20.9594 (1)	135.1023 (0)	108.4362 (3)	419.3822 (0)
	Median	10.7498 (0)	5.3584E-01 (0)	-21.9695 (1)	274.5813 (0)	27.0092 (3)	1004.9814 (0)
	Worst	13.3760 (0)	5.7998E-01 (0)	-22.6379 (1)	449.0876 (0)	247.3515 (4)	2673.6146 (0)
	c	0, 0, 0	0, 0, 0	0, 1, 1	0, 0, 0	3, 3, 3	0, 0, 0
	$\overline{v}$	0	0	9.5243E-01	0	4.4118	0
	Mean	10.7771	5.3096E-01	-21.9663	281.8723	120.0549	1156.2006
	Std	1.0967	3.1453E-02	2.1748	71.245	149.64	652.34
5×10^4	Best	3.5541E-02 (0)	4.7812E-02 (0)	-2.8572 (1)	8.1434E-03 (0)	3.2418E-01 (3)	7.2340E-05 (0)
	Median	7.0835E-02 (0)	1.1792E-01 (0)	-17.5667 (1)	3.5123E-02 (0)	-6.9395 (3)	7.2477E-04 (0)
	Worst	1.0435E-01 (0)	1.8054E-01 (0)	-17.7043 (1)	1.0796E-01 (0)	-10.2834 (3)	5.0123E-03 (0)
	c	0, 0, 0	0, 0, 0	0, 1, 1	0, 0, 0	0, 3, 3	0, 0, 0
	$\overline{v}$	0	0	8.0052E-01	0	2.9253E-01	0
	Mean	6.7359E-02	1.1745E-01	-17.0483	3.7329E-02	-2.3777	1.1996E-03
	Std	1.7622E-02	3.0024E-02	2.9605	2.1752E-02	11.933	1.3070E-03
5×10^5	Best	0 (0)	8.0700E-08 (0)	2.8000E-09 (0)	1.0000E-10 (0)	0 (0)	1.0000E-10 (0)
	Median	0 (0)	1.4314E-06 (0)	1.5890E-07 (0)	1.0000E-10 (0)	0 (0)	1.0000E-10 (0)
	Worst	0 (0)	2.5308E-02 (0)	7.4495E-06 (0)	1.0000E-10 (0)	0 (0)	1.0000E-10 (0)
	c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
	$\overline{v}$	0	0	0	0	0	0
	Mean	0	5.0731E-03	6.3957E-07	1.0000E-10	0	1.0000E-10
	Std	0	7.1385E-03	1.5396E-06	0	0	0

TABLE III

ERROR VALUES ACHIEVED WHEN FES= 5×10^3 , FES= 5×10^4, FES= 5×10^5 FOR PROBLEMS 7-12.

FES		g07	g08	g09	g10	g11	g12
5×10^3	Best	209.8858 (0)	1.3660E-05 (0)	51.6588 (0)	1332.7229 (3)	-7.4868E-01 (1)	7.7184E-05 (0)
	Median	567.2759 (0)	6.2579E-04 (0)	127.8131 (0)	4650.1063 (1)	-7.4935E-01 (1)	4.6174E-04 (0)
	Worst	326.1657 (1)	1.9840E-03 (0)	246.5905 (0)	1332.7229 (3)	-7.4948E-01 (1)	8.4977E-03 (0)
	c	0, 0, 0	0, 0, 0	0, 0, 0	0, 1, 1	0, 1, 1	0, 0, 0
	$\overline{v}$	0	0	0	0.0143	9.7907E-01	0
	Mean	799.4748	8.0252E-04	127.6818	9285.5880	-7.4908E-01	2.0888E-03
	Std	637.59	6.6687E-04	47.253	4150.5	1.4770E-03	3.1801E-03
5×10^4	Best	1.4241E-01 (0)	1.0000E-10 (0)	2.9610E-03 (0)	80.0974 (0)	-6.3099E-01 (1)	0 (0)
	Median	4.3237E-01 (0)	1.0000E-10 (0)	1.0509E-02 (0)	209.1018 (0)	-7.1068E-01 (1)	0 (0)
	Worst	9.4026E-01 (0)	1.0000E-10 (0)	3.6951E-02 (0)	384.4578 (0)	-7.1080E-01 (1)	0 (0)
	c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 1, 1	0, 0, 0
	$\overline{v}$	0	0	0	0	8.0207E-01	0
	Mean	4.6174E-01	1.0000E-10	1.2678E-02	214.7599	-7.0656E-01	0
	Std	1.7795E-01	0	8.3758E-03	86.930	1.6378E-02	0
5×10^5	Best	1.3232E-06 (0)	1.0000E-10 (0)	1.0000E-10 (0)	6.3843E-06 (0)	0 (0)	0 (0)
	Median	9.4367E-06 (0)	1.0000E-10 (0)	1.0000E-10 (0)	1.3432E-03 (0)	0 (0)	0 (0)
	Worst	1.1098E-04 (0)	1.0000E-10 (0)	1.0000E-10 (0)	2.5075E-01 (0)	0 (0)	0 (0)
	c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
	$\overline{v}$	0	0	0	0	0	0
	Mean	1.5699E-05	1.0000E-10	1.0000E-10	1.8911E-02	0	0
	Std	2.2678E-05	0	0	5.8279E-02	0	0

[5] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference On Neural Networks*, pages 1942–1948. IEEE, November 1995.

[6] J. Kennedy and R. Eberhart. *The Particle Swarm: Social Adaptation in Information-Processing Systems*. McGraw-Hill, London, 1999.

[7] R. Mendes, J. Kennedy, and J. Neves. The fully informed particle swarm: Simpler, maybe better. *IEEE Transactions on Evolutionary Computation*.

[8] E. Mezura. *Alternatives to Handle Constraints in Evolutionary Optimization*. PhD thesis, CINVESTAV-IPN, Mexico, DF, 2004.

[9] T. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*.

[10] R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical Report TR-95-012, International Computer Science Institute*.

[11] J. Zhang and F. Xie. Depso: Hybrid particle swarm with differential evolution operator. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 3816–3821. IEEE, October 2003.

TABLE IV

ERROR VALUES ACHIEVED WHEN FES= 5×10^3 , FES= 5×10^4, FES= 5×10^5 FOR PROBLEMS 13-18.

FES		g13	g14	g15	g16	g17	g18
	Best	3.1700E-01 (3)	-15.5490 (3)	3.6940E-02 (2)	1.6452E-01 (0)	-44.0775 (4)	7.0842E-01 (3)
	Median	2.6005E-01 (3)	-98.6435 (3)	-2.1374E-01 (2)	3.7704E-01 (0)	717.5972 (4)	6.1333E-01 (8)
	Worst	1.6674E-01 (3)	-150.2403 (3)	-1.0318 (2)	4.7513E-01 (0)	-743.0372 (4)	4.5901E-01 (9)
5×10^3	c	0, 3, 3	3, 3, 3	0, 2, 2	0, 0, 0	4, 4, 4	3, 8, 8
	$\overline{v}$	5.0549E-01	2.9882	6.4033E-01	0	7.1644	5.9422E-01
	Mean	2.0622E-01	-94.3657	-1.1302E-01	3.4332E-01	59.7829	6.8096E-01
	Std	1.8175E-01	32.996	7.1683E-01	9.8220E-02	515.54	5.9788E-01
	Best	-3.6754E-02 (3)	-24.2074 (3)	-1.1098 (2)	3.9033E-05 (0)	31.1470 (4)	7.3575E-03 (0)
	Median	-3.7477E-02 (3)	-26.8854 (3)	-1.1616 (2)	9.1508E-05 (0)	-26.2083 (4)	2.2331E-02 (0)
	Worst	-3.8119E-02 (3)	-27.6668 (3)	-1.1854 (2)	2.0331E-04 (0)	41.9462 (4)	2.0023E-01 (0)
5×10^4	c	0, 3, 3	0, 3, 3	0, 2, 2	0, 0, 0	0, 4, 4	0, 0, 0
	$\overline{v}$	7.0527E-01	7.7558E-01	7.5506E-01	0	5.7395E-01	0
	Mean	-3.7217E-02	-26.5212	-1.1521	9.5106E-05	-6.1823	3.9529E-02
	Std	1.3740E-03	1.0130	2.7829E-02	4.0146E-05	29.184	4.4179E-02
	Best	0 (0)	1.8753E-04 (0)	1.0000E-10 (0)	1.0000E-10 (0)	1.7480E-01 (0)	0 (0)
	Median	1.1200E-08 (0)	3.2912E-03 (0)	1.0000E-10 (0)	1.0000E-10 (0)	13.9638 (0)	4.0000E-10 (0)
	Worst	2.6898E-05 (0)	1.2560E-01 (0)	1.0000E-10 (0)	1.0000E-10 (0)	110.5035 (0)	1.9105E-01 (0)
5×10^5	c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
	$\overline{v}$	0	0	0	0	0	0
	Mean	2.2650E-06	1.5919E-02	1.0000E-10	1.0000E-10	40.0393	7.7974E-03
	Std	6.3801E-06	2.8553E-02	0	0	42.286	3.8184E-02

TABLE V

ERROR VALUES ACHIEVED WHEN FES= 5×10^3 , FES= 5×10^4, FES= 5×10^5 FOR PROBLEMS 19-24.

FES		g19	g20	g21	g22	g23	g24
	Best	423.5117 (0)	5.9014 (20)	346.0798 (5)	-234.2854 (20)	36.7985 (5)	8.8957E-03 (0)
	Median	680.4383 (0)	7.3553 (20)	526.6707 (5)	153.3767 (20)	183.7565 (5)	3.5427E-02 (0)
	Worst	1012.5704 (0)	8.1033 (20)	764.0398 (5)	1085.8097 (19)	254.5868 (5)	7.3748E-02 (0)
5×10^3	c	0, 0, 0	2,19,20	3, 5, 5	19,20,20	2, 5, 5	0, 0, 0
	$\overline{v}$	0	3.7490	1.3027	3431690.2080	8.9800E-01	0
	Mean	681.0290	7.0074	424.5241	2284.4529	49.7630	3.2966E-02
	Std	149.31	1.4436	156.33	4000.7	301.30	1.6321E-02
	Best	5.5783 (0)	6.2618E-02 (20)	70.7688 (5)	333.6749 (19)	-204.9999 (4)	1.0000E-10 (0)
	Median	7.9308 (0)	6.5519E-02 (20)	-12.9154 (4)	997.8696 (20)	-59.2799 (4)	1.6000E-09 (0)
	Worst	13.3192 (0)	1.9984E-01 (20)	70.9991 (5)	545.0811 (20)	-42.0698 (4)	4.8000E-09 (0)
5×10^4	c	0, 0, 0	0, 8,20	0, 5, 5	19,19,19	0, 4, 4	0, 0, 0
	$\overline{v}$	0	1.0258E-01	2.4101E-01	137986.7000	3.7082E-01	0
	Mean	8.7535	1.4672E-01	40.4728	1440.1355	-54.8925	1.7760E-09
	Std	2.0777	6.8771E-02	56.869	2193.2	91.663	1.2347E-09
	Best	8.7788E-04 (0)	1.3460E-02 (8)	1.2713E-01 (0)	342.7683 (19)	46.8846 (0)	0 (0)
	Median	2.6302E-02 (0)	3.2600E-02 (8)	81.3460 (0)	14198.8059 (19)	130.5043 (0)	0 (0)
	Worst	6.8308E-01 (0)	1.8513E-02 (16)	325.9271 (0)	16539.3498 (19)	117.5070 (4)	0 (0)
5×10^5	c	0, 0, 0	0, 5, 6	0, 0, 0	16,19,19	0, 0, 0	0, 0, 0
	$\overline{v}$	0	1.3148E-01	0	4.1939	0	0
	Mean	9.4118E-02	2.9980E-02	78.7134	3620.6318	162.2006	0
	Std	1.6158E-01	4.9951E-03	67.019	4963.7	87.634	0

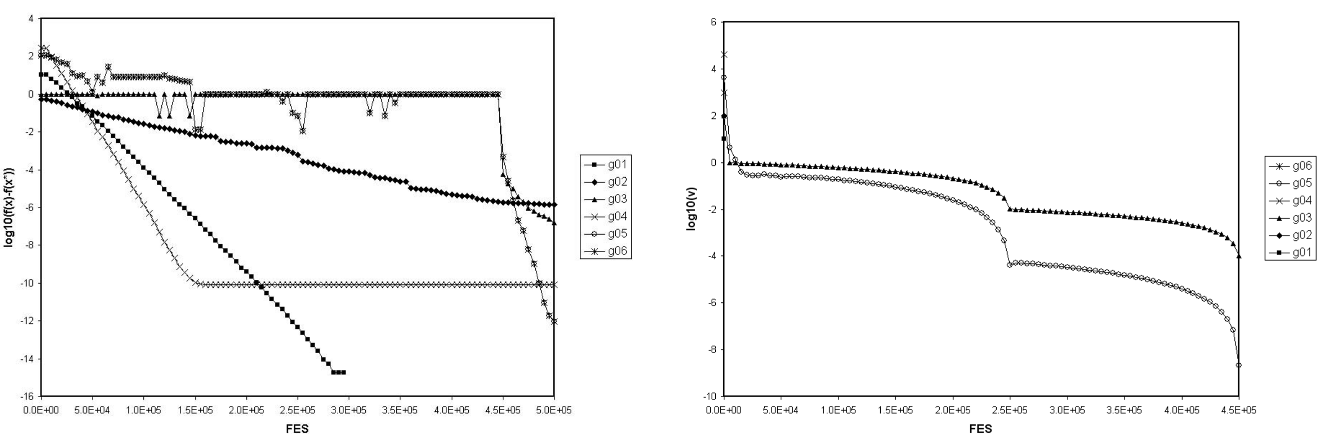


Fig. 8. Convergence Graph for Problems 1-6

TABLE VI

NUMBER OF FES TO ACHIEVE THE FIXED ACCURACY LEVEL $((f(\vec{x}) - f(\vec{x}*)) \leq 0.0001)$, SUCCESS RATE, FEASIBLE RATE AND SUCCESS PERFORMANCE.

Prob.	Best	Median	Worst	Mean	Std	Feasible Rate	Success Rate	Success Performance
g01	95100	102100	106900	101532	3072	100.00%	100.00%	101532
g02	180000	219400	327900	231193	42473	100.00%	56.00%	412844.3878
g03	450100	450100	454000	450644	1084	100.00%	100.00%	450644
g04	74300	79300	85000	79876	2692	100.00%	100.00%	79876
g05	450100	452300	457200	452256	1534	100.00%	100.00%	452256
g06	47800	56800	61100	56508	3922	100.00%	100.00%	56508
g07	198600	358600	444100	352592	58266	100.00%	96.00%	367282.9861
g08	2800	6100	8400	6124	1329	100.00%	100.00%	6124
g09	77000	96400	129000	97544	11997	100.00%	100.00%	97544
g10	398000	468350	475600	452575	36678	100.00%	16.00%	2828593.75
g11	450100	450100	450100	450100	0	100.00%	100.00%	450100
g12	3300	8100	10900	8088	1899	100.00%	100.00%	8088
g13	450100	450100	453200	450420	812	100.00%	100.00%	450420
g14	NA	NA	NA	NA	NA	100.00%	0.00%	NA
g15	450100	450100	450100	450100	0	100.00%	100.00%	450100
g16	43400	48700	53900	49040	2402	100.00%	100.00%	49040
g17	NA	NA	NA	NA	NA	100.00%	0.00%	NA
g18	120800	211800	394900	214322	78252	100.00%	92.00%	232958.4121
g19	NA	NA	NA	NA	NA	100.00%	0.00%	NA
g20	NA	NA	NA	NA	NA	0.00%	0.00%	NA
g21	NA	NA	NA	NA	NA	100.00%	0.00%	NA
g22	NA	NA	NA	NA	NA	0.00%	0.00%	NA
g23	NA	NA	NA	NA	NA	96.00%	0.00%	NA
g24	14600	19900	22700	19980	2008	100.00%	100.00%	19980

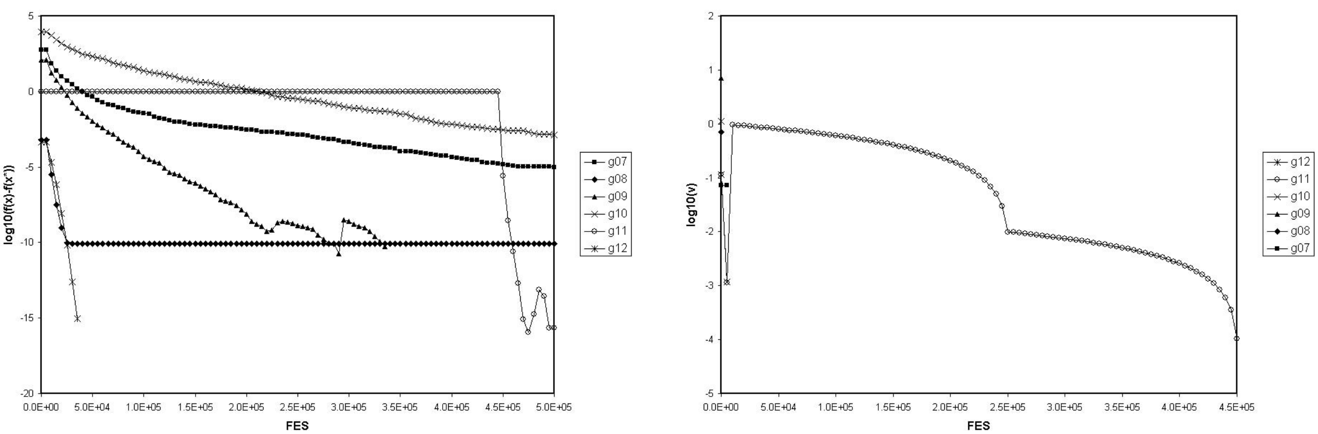


Fig. 9. Convergence Graph for Problems 7-12

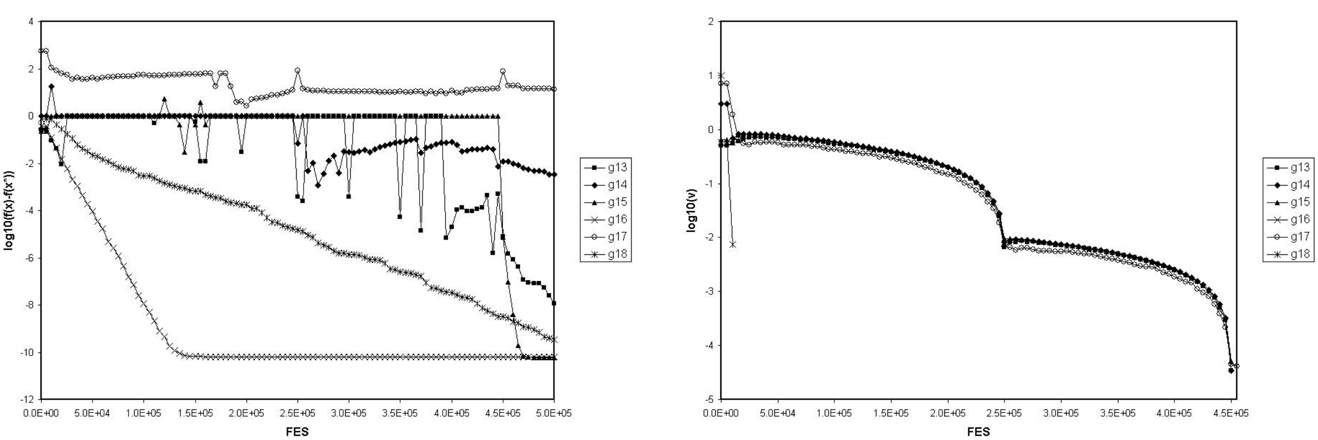


Fig. 10. Convergence Graph for Problems 13-18

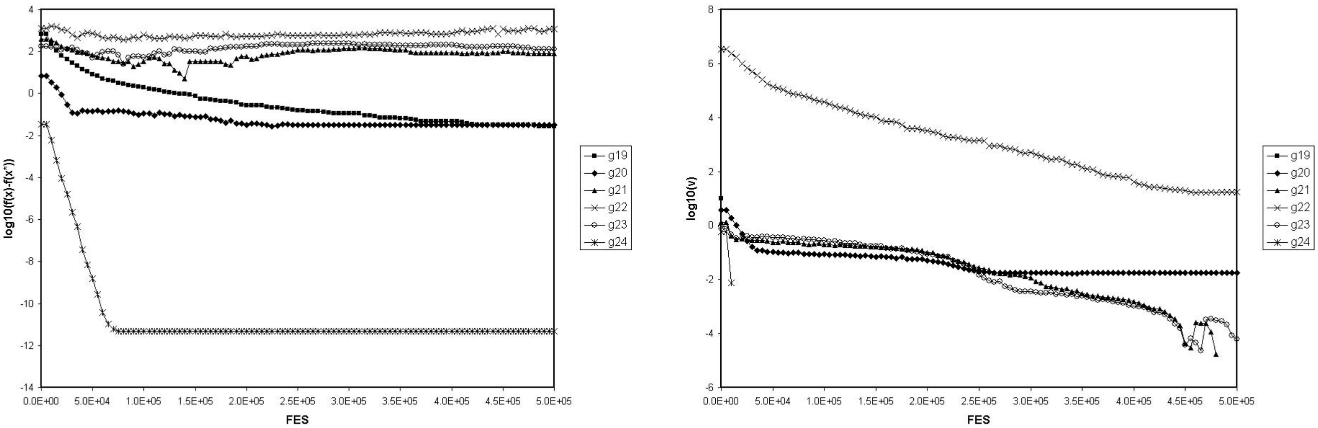


Fig. 11. Convergence Graph for Problems 19-24

TABLE VII
THE RESULTS OF PESO+ IN THE WELL KNOWN FORMAT USED BY THE COMMUNITY

TF		Optimal	Best	Median	Mean	Worst	Standard Deviation	Feasible Solutions
g01	Min	-15.0000000000	-15.000000	-15.000000	-15.000000	-15.000000	0	30/30
g02	Min	-0.8036191042	-0.803619	-0.803616	-0.800062	-0.785266	5.0599E-03	30/30
g03	Min	-1.0005001000	-1.000500	-1.000500	-1.000500	-1.000499	3.3805E-07	30/30
g04	Min	-30665.5386717834	-30665.538672	-30665.538672	-30665.538672	-30665.538672	0	30/30
g05	Min	5126.4967140071	5126.496714	5126.496714	5126.496714	5126.496714	0	30/30
g06	Min	-6961.8138755802	-6961.813876	-6961.813876	-6961.813876	-6961.813876	0	30/30
g07	Min	24.3062090681	24.306209	24.306214	24.306223	24.306301	2.2432E-05	30/30
g08	Min	-0.0958250415	-0.095825	-0.095825	-0.095825	-0.095825	0	30/30
g09	Min	680.6300573745	680.630057	680.630057	680.630057	680.630057	0	30/30
g10	Min	7049.2480205286	7049.248027	7049.250013	7049.262277	7049.349764	2.6841E-02	30/30
g11	Min	0.7499000000	0.749999	0.749999	0.749999	0.749999	0	30/30
g12	Min	-1.0000000000	-1.000000	-1.000000	-1.000000	-1.000000	0	30/30
g13	Min	0.0539415140	0.053942	0.053942	0.053946	0.054022	1.6856E-05	30/30
g14	Min	-47.7648884595	-47.764793	-47.762253	-47.759266	-47.744179	6.7697E-03	30/30
g15	Min	961.7150222899	961.715171	961.715171	961.715171	961.715171	0	30/30
g16	Min	-1.9051552586	1.905155	1.905155	1.905155	1.905155	0	30/30
g17	Min	8853.5396748064	8853.714472	8866.110257	8890.491657	8966.886811	37.8748	30/30
g18	Min	-0.8660254038	-0.866025	-0.866025	-0.850638	-0.674549	5.2932E-02	30/30
g19	Min	32.6555929502	32.658545	32.701711	32.762097	33.281978	1.6008E-01	30/30
g20	Min	0.2049794002	-	-	-	-	-	0/30
g21	Min	193.7245100700	193.851635	261.043846	276.792683	708.140549	96.4288	30/30
g22	Min	236.4309755040	-	-	-	-	-	0/30
g23	Min	-400.0551000000	-353.170481	-219.386636	-183.151218	121.107212	129.6548	30/30
g24	Min	-5.5080132716	-5.508013	-5.508013	-5.508013	-5.508013	0	30/30